Available online at www.sciencedirect.com
ScienceDirect
Science of Computer Programming 66 (2007) 1–3
Science of Computer Programming
www.elsevier.com/locate/scico

Bridging the gap between enterprise architectures and software architectures

The US Federal Agencies' Chief Information Officer Council in its guide to Federal Enterprise Architecture has defined an Enterprise Architecture (EA) to be "a strategic information asset base, which defines the mission, the information necessary to perform the mission and the technologies necessary to perform the mission, and the transitional processes for implementing new technologies in response to changing mission needs". An EA is an architecture, and hence by definition the EA serves as the "blueprint" for all information systems developed in an organization. In other words, we may consider the EA as capturing the information technology architecture of an organization including hardware, software, and networking standardizations, if any, that serves as the basis for all information systems developed within an organization. This viewpoint enables us to put EA in proper business perspective: the EA is derived from, and closely aligned with, the Strategic Enterprise Plan (SEP) of the organization where the SEP captures the high level business objectives and goals for the organization for the next 3 to 5 years. The Strategic Information Systems Plan (SISP), usually developed by the Information Systems business unit of the organization, is derived from the SEP, and prioritizes the information systems development projects that will be undertaken in the near term (next 3 to 5 years) by the organization that will help achieve the business goals of the organization set in the SEP. Any information system that is approved for development by the SISP usually, after another round of approvals by the executive sponsors, goes through the typical system development process whose initial phases include scope definition, problem analysis and requirements analysis.

During the requirements analysis phase the requirements of the new system are elicited from the stakeholders and analyzed — the analysis includes the development of system architectures (SysA) which considers allocation of the requirements between hardware, software and the network. Selection of the optimal SA from among the candidate system architectures is an important part of system development, since the quality of the system architecture has an effect on the quality of the final system — generally, a high quality system architecture results in a high quality final system. Usually the system requirements drive the selection of the system architecture from among competing architectures — but this approach ignores the effects of the EA on SysA that ensures that the system architecture meets the requirements set by the EA, such as satisfying the requirements of the chosen enterprise technology. Therefore, it will be extremely useful to the organization to have an understanding of the extent to which the SysA satisfies the EA — this *traceability* will ensure that the chosen system architecture meets the goals of the enterprise architecture as well.

Once the optimal system architecture is selected, the next step is to develop the hardware, software and the network elements of the system almost independently of each other. Among these elements, software development is usually the most important, since the development of software seems to consume most resources in terms of budget, schedule, and manpower. The requirements pertaining to software are used to design the software subsystem, and the first step in the design is usually the development of the software architecture (SwA) for the system. The SwA is the high level viewpoint of the software to be developed and consists of components, connectors, constraints, styles, patterns, and the like. Most of the properties of the final software are considered to have been designed into the software at the software architectural design stage itself, hence, in general, high quality software architecture being expected to result in the development of a high quality software system. Like for the SysA, SwA selection is driven not only by the software requirements but also by the SysA, and since the SysA is influenced by the EA, we can see that the SwA is influenced by the EA as well.

0167-6423/$ - see front matter
doi:10.1016/j.scico.2006.12.001

Ensuring traceability between the SwA and the EA will help ensure that the SwA for the project satisfies the enterprise's information infrastructure, thereby helping achieve the enterprise's business goals. While this relationship can be appreciated, it may seem tenuous — however, it is important for the software industry practitioners and researchers that this tenuous link be explored seriously so that better techniques are developed to ensure the core artifact of all information systems developed, namely, the SwA, is indeed aligned with, and derives its strengths from, the EA. Studies have shown that such an alignment actually increases the profitability of the enterprise.

The theme of *the 5th International Workshop on System/Software Architectures (IWSSA'06)* held in Las Vegas in June 2006, was *bridging the gap between the enterprise architecture and the (potentially distributed) software architecture*. The papers presented at this workshop explored several techniques for ensuring alignment between the EA and SysA. IWSSA'06 was held as part of International Conference on Software Engineering Research and Practice, 2006. On the basis of the audience response and the quality of papers attracted, we feel that the workshop was successful in meeting its objective of bringing together a diverse group of academicians and industry practitioners interested in the workshop's theme. Authors of selected papers among those presented at the workshop were invited to submit extended and revised versions of their papers for consideration for this special issue and the final six papers selected for this special issue were approved by the Program Committee members after two rounds of rigorous reviews.

The first paper, entitled "Ontology-driven middleware for next-generation train backbones", by Stijn Verstichel, Sofie Van Hoecke, Matthias Strobbe, Steven Van den Berghe, Filip De Turck, Bart Dhoedt, Piet Demeester, and Frederik Vermeulen, all from Belgium, proposes a novel approach for developing middleware for integrating various information systems for railway management — the EA is that of the railways as a whole while the SwA is considered for the system-integration-enabling middleware software. This paper uses the ontology of the railway domain to capture the essential concepts to provide the intelligence needed for integration.

The second paper, entitled "Definition and Use of Computation Independent Models in an MDA-Based Groupware Development Process", by José Luis Garrido, Manuel Noguera, Miguel González, María V. Hurtado, and María L. Rodríguez, all from Spain, uses the concept of ontology to model CSCW (Computer-Supported Cooperative Work) systems. Here the ontology is used to capture the organization of the enterprise and the UML is used to model the groupware system — together they help achieve model-driven architecture (MDA) for the system. CSCW may be viewed as another form of middleware but in a different sense — CSCW integrate applications across businesses.

The third paper, entitled "An Architecture for Access Control Management in Collaborative Enterprise Systems Based on Organization Models", by F.L. Gutiérrez Vela, J.L. Isla Montes, P. Paderewski Rodríguez, M. Sánchez Román, and B. Jiménez Valverde, again all from Spain, develops service-oriented architectures that model organizations in order to develop access control management systems. Pattern definition languages are used to model the EA requirements for the management systems.

The fourth paper, entitled "Software Requirements and Architecture Modeling for Evolving Non-Secure Applications into Secure Applications", by Michael Shin and Hassan Gomaa, both from the US, describes evolution of a non-secure application into a secure application. Usually the EA specifies constraints in information systems to be developed but security may be a requirement in the later generations of software — therefore, non-secure software may need to be made secure and this paper uses the separation of concerns approach to improve security.

The fifth paper, entitled "Designing and Managing Evolving Systems using a MAS-Product-Line Approach" by Joaquin Peña, Michael G. Hinchey, Manuel Resinas, Roy Sterritt, and James L. Rash, including researchers from Spain, UK, and the US, models the EA as a multi-agent system with components of the system represented by agents. The MaCMAS (Methodology for Analyzing Complex Multiagent Systems) methodology is proposed for modeling the architectures for product-lines.

The sixth and the final paper, entitled "Using FDAF to Bridge the Gap Between Enterprise and Software Architectures for Security", by Lirong Dai and Kendra Cooper, both from the US, provides a systematic framework called the Formal Design Analysis Framework (FDAF) that can help bridge the gap between the SysA and SwA. The application of FDAF is illustrated by achieving security goals of the enterprise in an on-line banking system.

We hope the readers enjoy the papers in this special issue on the important topic of bridging the gap between enterprise architectures and software architectures. We would like to thank all the participants of IWSSA'06 for their enthusiastic reception and energetic critique of the papers. We are extremely appreciative of the effort of the Program Committee for IWSSA'06 for their thorough and excellent reviews that helped us to short-list the final six high quality papers for this special issue. Members of the Program Committee included Dr. Philippe Aniorte, LIUPPA IUT de Bayonne, France; Dr. Doo-Hwan Bae, KAIST, Korea; Dr. Roger Champagne, Ecole de Technologie Superieure,

Canada; Dr. Francois Coallier, Ecole de Technologie Superieure, Canada; Dr. Kendra Cooper, The University of Texas at Dallas, USA; Dr. Lirong Dai, Seattle University, USA; Dr. Sergiu Dascalu, University of Nevada, Reno, USA; Dr. Lars Grunske, University of Queensland, Australia; Dr. Fred Harris, University of Nevada, Reno, USA; Dr. Michael Hinchey, NASA, USA; Dr. Stan Jarzabek, National University of Singapore, Singapore; Dr. Carlos Juiz, University of the Balearic Islands, Spain; Dr. Rick Kazman, University of Hawaii and Software Engineering Institute, USA; Dr. Chung-Horng Lung, Carleton University, Canada; Dr. Pericles Loucopoulos, The University of Manchester, UK; Dr. María Dolores Lozano, University of Castilla-La Mancha, Spain; Dr. Murakami Masaki, Okayama University, Japan; Dr. Tommi Mikkonen, Tampere University of Technology, Finland; Dr. Patricia Paderewski, University of Granada, Spain; Dr. Sooyong Park, Sogang University, Korea; Dr. Juan Ramil, Open University, UK; Vespe Savikko, Elektrobit, Finland; Dr. Motoshi Saeki, Tokyo Institute of Technology, Japan; Dr. Michael Shin, Texas Tech University, USA; Dr. Yeong-Tae Song, Towson University, USA; and Dr. Francisco Luis Gutierrez Vela, University of Granada, Spain. We would like to thank Dr. Hamid Arabnia, the organizer of the multiconference at Las Vegas, for the excellent support extended to us. Finally we would like to thank Dr. Jan Bergstra, Journal of Science of Computer Programming, for agreeing to release this special issue, and Dr. Bas van Vlijmen, University of Amsterdam, for providing encouraging support throughout the preparation of this special issue.

Lawrence Chung*
Department of Computer Science,
University of Texas at Dallas,
Richardson, TX 75083,
United States
E-mail address: chung@utdallas.edu.

Nary Subramanian
Department of Computer Science,
University of Texas at Tyler,
Tyler, TX 75799,
United States

Available online 25 January 2007

* Corresponding editor.